



Attorney's Docket No.: 07844-501001 / P464

2151  
22/2

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : R. Perelman, et al.                      Art Unit : 2151  
Serial No. : 10/006,260                              Examiner : Backhean Tiv  
Filed : November 2, 2001  
Title : CLIENT-SIDE MODIFICATION OF ELECTRONIC DOCUMENTS IN A  
CLIENT-SERVER ENVIRONMENT

**Mail Stop Appeal Brief - Patents**  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

BRIEF ON APPEAL

Sir:

Applicant herewith files this brief on appeal under 37 CFR 41.37, thereby perfecting the notice of appeal which was originally filed on November 1, 2005.

The sections required by 37 CFR 41.37 follow.

**(1) Real Party in Interest**

This application is assigned of record to Adobe Systems Incorporated who is hence the real party in interest.

**(2) Related Appeals and Interferences**

There are no known related appeals or interferences.

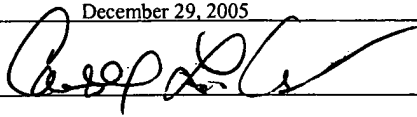
01/04/2006 BABRAHA1 00000079 10006260

01 FC:1402

500.00 OP

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

I hereby certify under 37 CFR §1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Date of Deposit December 29, 2005  
Signature 

Carroll Allman  
Typed or Printed Name of Person Signing Certificate

**(3) Status of Claims**

Claims 1-40 are pending in the application and stand rejected, and all of these claims are appealed herein.

**(4) Status of Amendments**

No claim amendments have been filed after final rejection.

**(5) Summary of Claimed Subject Matter**

The presently claimed subject matter is generally directed to client-side modification of electronic documents by server-generated instructions in a client-server environment. As briefly summarized in paragraph 12 of the present application, "Client-side modification of electronic documents by server-generated instructions in a client-server environment enables dynamic modification of formatting information in an electronic document to accommodate new data received from the server. A server generates machine instructions to send to a client along with new data for an electronic document. The generated instructions modify the electronic document at the client to accommodate the new data." (*See* ¶ 12 of the Specification at page 5, lines 8-19.)

Independent claims 1, 16, 24 and 32 cover this subject matter broadly. Independent claim 1 includes receiving a request from a client (*see e.g.*, ¶ 33 of the Specification at page 10, lines 5-14, reference numeral 160 in Fig. 1B); producing data corresponding to the client request (*see e.g.*, ¶ 33 of the Specification at page 10, lines 5-14, reference numeral 162 in Fig. 1B; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23; and Table 1); generating

instructions to modify an electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to present the electronic document, the generated instructions specifying one or more operations to modify the electronic document's predetermined format at the client to accommodate the produced data, the generated instructions to be performed at the client to effect the one or more operations (*see e.g.*, ¶ 33 of the Specification at page 10, lines 5-14, reference numeral 164 in Fig. 1B; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23; and Table 1); and transmitting the produced data and the generated instructions to the client (*see e.g.*, ¶ 34 of the Specification at page 10, lines 14-21, reference numeral 166 in Fig. 1B; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23; and Table 1).

Dependent claim 4 further specifies that the operations to modify the predetermined format at the client comprise adding information to the electronic document without changing pre-existing information in the electronic document (*see e.g.*, ¶s 37-38 of the Specification at page 11, line 14 to page 12, line 5; ¶ 47 of the Specification at page 15, lines 4-7; and ¶ 58 of the Specification at page 24, lines 13-22, reference numerals 430 and 435 in FIGs. 4A and 4B).

Dependent claim 6 further specifies that the generated instructions further comprise at least one tag indicating an order in which the produced data is to be imported into the electronic document and the instructions are to be performed (*see e.g.*, ¶ 39 of the Specification at page 12, lines 6-19; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23, reference numerals 236 and 250 in FIGs. 2B and 2C; and Table 1).

Dependent claim 7 further specifies that the at least one tag is a before tag, an after tag or a document tag, wherein the before tag indicates that the produced data is to be imported into the

electronic document before the instructions are performed, the after tag indicates that the produced data is to be imported into the electronic document after the instructions are performed, and the document tag indicates that at least a portion of the generated instructions are to be inserted into the electronic document (*see e.g.*, ¶ 39 of the Specification at page 12, lines 6-19; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23, reference numerals 236 and 250 in FIGs. 2B and 2C; and Table 1).

Dependent claim 9 further specifies that adding the information causes a new visual object to overlay one or more pre-existing visual objects in the electronic document (*see e.g.*, ¶ 37 of the Specification at page 11, line 14 to page 12, line 2; ¶ 54 of the Specification at page 17, lines 14-19; and ¶ 58 of the Specification at page 24, lines 13-22, reference numerals 430 and 435 in FIGs. 4A and 4B).

Independent claim 16 includes obtaining an electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to present the electronic document (*see e.g.*, ¶ 31 of the Specification at page 9, lines 13-20, reference numeral 150 in Fig. 1B); sending a request to a server (*see e.g.*, ¶ 32 of the Specification at page 9, line 21 to page 10, line 4, reference numeral 155 in Fig. 1B); receiving, from the server, data and instructions to modify the electronic document (*see e.g.*, ¶ 35 of the Specification at page 10, line 22 to page 11, line 6, reference numeral 170 in Fig. 1B); and modifying the electronic document's predetermined format in accordance with the received instructions to accommodate the received data (*see e.g.*, ¶ 35 of the Specification at page 10, line 22 to page 11, line 6, reference numeral 175 in Fig. 1B; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23; and Table 1).

Dependent claim 19 further specifies that modifying the electronic document's predetermined format comprises adding information to the electronic document without changing pre-existing information in the electronic document (*see e.g.*, ¶s 37-38 of the Specification at page 11, line 14 to page 12, line 5; ¶ 47 of the Specification at page 15, lines 4-7; and ¶ 58 of the Specification at page 24, lines 13-22, reference numerals 430 and 435 in FIGs. 4A and 4B).

Dependent claim 21 further specifies that the instructions further comprise at least one tag indicating an order in which the produced data is to be imported into the electronic document and the instructions are to be performed (*see e.g.*, ¶ 39 of the Specification at page 12, lines 6-19; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23, reference numerals 236 and 250 in FIGs. 2B and 2C; and Table 1).

Dependent claim 22 further specifies that the at least one tag is a before tag, an after tag or a document tag, wherein the before tag indicates that the produced data is to be imported into the electronic document before the instructions are performed, the after tag indicates that the produced data is to be imported into the electronic document after the instructions are performed, and the document tag indicates that at least a portion of the generated instructions are to be inserted into the electronic document (*see e.g.*, ¶ 39 of the Specification at page 12, lines 6-19; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23, reference numerals 236 and 250 in FIGs. 2B and 2C; and Table 1).

Independent claim 24 includes receiving a request from a client (*see e.g.*, ¶ 33 of the Specification at page 10, lines 5-14, reference numeral 160 in Fig. 1B); producing data corresponding to the client request (*see e.g.*, ¶ 33 of the Specification at page 10, lines 5-14,

reference numeral 162 in Fig. 1B; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23; and Table 1); generating instructions to modify an electronic document defining visual information to be displayed, the generated instructions specifying one or more operations to modify the electronic document at the client to accommodate the produced data, and the generated instructions including at least one tag indicating an order in which the produced data is to be imported into the electronic document and the instructions are to be performed , and the generated instructions to be performed at the client to effect the one or more operations (*see e.g.*, ¶ 33 of the Specification at page 10, lines 5-14, reference numeral 164 in Fig. 1B; ¶ 39 of the Specification at page 12, lines 6-19; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23, reference numerals 236 and 250 in FIGs. 2B and 2C; and Table 1); and transmitting the produced data and the generated instructions to the client (*see e.g.*, ¶ 34 of the Specification at page 10, lines 14-21, reference numeral 166 in Fig. 1B; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23; and Table 1).

Dependent claim 25 further specifies that the at least one tag indicates that the produced data is to be imported into the electronic document before the instructions are performed (*see e.g.*, ¶ 39 of the Specification at page 12, lines 6-19; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23, reference numeral 236 in Fig. 2B; and Table 1).

Dependent claim 26 further specifies that the at least one tag indicates that the produced data is to be imported into the electronic document after the instructions are performed (*see e.g.*, ¶ 39 of the Specification at page 12, lines 6-19; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23, reference numeral 250 in Fig. 2C; and Table 1).

Dependent claim 27 further specifies that the at least one tag indicates that at least a portion of the generated instructions are to be inserted into the electronic document (*see e.g.*, ¶ 39 of the Specification at page 12, lines 6-19; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23; and Table 1).

Dependent claim 29 further specifies that the operations to modify the electronic document comprise adding formatting information to the electronic document without changing pre-existing formatting information in the electronic document (*see e.g.*, ¶s 37-38 of the Specification at page 11, line 14 to page 12, line 5; ¶ 47 of the Specification at page 15, lines 4-7; and ¶ 58 of the Specification at page 24, lines 13-22, reference numerals 430 and 435 in FIGs. 4A and 4B).

Independent claim 32 includes receiving a request from a client (*see e.g.*, ¶ 33 of the Specification at page 10, lines 5-14, reference numeral 160 in Fig. 1B); producing data corresponding to the client request (*see e.g.*, ¶ 33 of the Specification at page 10, lines 5-14, reference numeral 162 in Fig. 1B; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23; and Table 1); generating instructions for an electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to present the electronic document, the generated instructions specifying one or more operations to modify the predetermined format at the client to accommodate the produced data, the generated instructions to be performed at the client to effect the one or more operations (*see e.g.*, ¶ 33 of the Specification at page 10, lines 5-14, reference numeral 164 in Fig. 1B; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23; and Table 1); and transmitting the produced data and the generated instructions to the client (*see e.g.*, ¶ 34 of the Specification at page 10,

lines 14-21, reference numeral 166 in Fig. 1B; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23; and Table 1). (Emphasis added.)

Dependent claim 35 further specifies that the operations to modify the predetermined format at the client comprise adding information to the electronic document without changing pre-existing information in the electronic document (*see e.g.*, ¶s 37-38 of the Specification at page 11, line 14 to page 12, line 5; ¶ 47 of the Specification at page 15, lines 4-7; and ¶ 58 of the Specification at page 24, lines 13-22, reference numerals 430 and 435 in FIGs. 4A and 4B).

Dependent claim 37 further specifies that the generated instructions further comprise at least one tag indicating an order in which the produced data is to be imported into the electronic document and the instructions are to be performed (*see e.g.*, ¶ 39 of the Specification at page 12, lines 6-19; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23, reference numerals 236 and 250 in FIGs. 2B and 2C; and Table 1).

Dependent claim 38 further specifies that the at least one tag is a before tag, an after tag or a document tag, wherein the before tag indicates that the produced data is to be imported into the electronic document before the instructions are performed, the after tag indicates that the produced data is to be imported into the electronic document after the instructions are performed, and the document tag indicates that at least a portion of the generated instructions are to be inserted into the electronic document (*see e.g.*, ¶ 39 of the Specification at page 12, lines 6-19; ¶s 44-55 of the Specification at page 14, line 4 to page 17, line 23, reference numerals 236 and 250 in FIGs. 2B and 2C; and Table 1).

Dependent claim 40 further specifies that adding the information causes a new visual object to overlay one or more pre-existing visual objects in the electronic document (*see e.g.*,



¶ 37 of the Specification at page 11, line 14 to page 12, line 2; ¶ 54 of the Specification at page 17, lines 14-19; and ¶ 58 of the Specification at page 24, lines 13-22, reference numerals 430 and 435 in FIGs. 4A and 4B).

**(6) Grounds of Rejection to be Reviewed on Appeal**

I. Claims 1-40 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over D'Arlach et al. (US 6,026,433) in view of Szabo (US 6,868,525).

II. Claims 1-40 also stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over D'Arlach et al. in view of Dilworth et al. (WO 00/51018).

**(7) Argument**

**Ground of Rejection I - Claims 1-40**

**Independent Claims 1, 16, 24 and 32 – Client-Side Modification**

D'Arlach describes a "method for creating and editing a Web site in a client-server computer network using customizable templates." (See D'Arlach at Abstract.) But D'Arlach does not describe generating instructions to modify an electronic document, the generated instructions specifying one or more operations to modify the electronic document at the client to accommodate the produced data, and the generated instructions to be performed at the client to effect the one or more operations. Nowhere does D'Arlach describe or show generation of instructions to be performed at the client to effect the one or more operations, as claimed. D'Arlach clearly describes a system in which a client sends requests to a server, which generates documents to be sent to the client in the course of building a Web site using customizable

templates. Copies of the template documents are changed in response to user requests, but D'Arlach makes very clear that all such changes occur on the server computer, not at the client. (See D'Arlach at col. 5, lines 26-32.) Thus, D'Arlach actually teaches away from generating instructions to send to a client to modify an electronic document at the client.

The Advisory Action mailed October 6, 2005 fails to address this point, but rather states, "D'Arlach teaches that a user can modify a template. The template has instructions for modification." This statement fails to address the claim limitations at issue and also fails to address the cited portion of D'Arlach (col. 5, lines 26-32), which teaches away from the presently claimed subject matter. As briefly summarized in paragraph 12 of the present application, "Client-side modification of electronic documents by server-generated instructions in a client-server environment enables dynamic modification of formatting information in an electronic document to accommodate new data received from the server. A server generates machine instructions to send to a client along with new data for an electronic document. The generated instructions modify the electronic document at the client to accommodate the new data." In stark contrast, D'Arlach teaches the use of customizable templates to edit a Web site in a client-server computer network, where a client sends requests to the server to make changes to the Web site, and the server makes the changes at the server. Thus, D'Arlach fails to teach or suggest generating instructions to send to a client to modify an electronic document at the client to accommodate produced data.

For at least the above reasons, all of claims 1-40 are patentable over D'Arlach, Szabo and Dilworth, and these claims should be allowed.

### **Independent Claims 1, 16 and 32 – Predetermined Format**

With respect to Szabo, the Examiner has acknowledged that D'Arlach does not describe an electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to present the electronic document, and relies on Szabo for this feature.

Szabo describes a “human user computer interface system, providing a graphic representation of a hierarchy populated with naturally classified objects, having included therein at least one associated object having a distinct classification.” (*See Szabo at Abstract.*) The Examiner contends that Szabo teaches the use of XML (Extensible Markup Language) as an electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to present the electronic document. Szabo does not in any way teach or suggest this. In particular, the cited portion of Szabo (col. 2, lines 15-30) cannot support the Examiner's assertion that, “an XML document will appear the same no matter what platform the OS is”, for this portion of Szabo says nothing about the appearance of an XML document on different platforms, and in fact, an XML document will appear very differently on different platforms depending on the software used to open and display the XML document.

Moreover, the Examiner's statement that, “an XML document will appear the same no matter what platform the OS is” does not actually address the claimed feature. The claims recite an electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to present the electronic document. The format of XML does not define document appearance at all. XML is by definition a general-purpose markup language that can be rendered differently in different Web browsers.

As described in paragraphs 9 and 10 of the present application, the claimed electronic document is generally a type of document that allows a publisher to control the look and feel of the document as seen by an end user, including the specific physical page or pages on which information appears when printed or displayed. Thus, the format should generally support and preserve all visual formatting features (e.g., fonts, graphics, color, etc.) of any source document, regardless of the source computer platform and/or software application used to create the source document. The ability to control final appearance, or look-and-feel, of an electronic document as viewed by a reader can be a critical branding issue for businesses and other publishing organizations, and is particularly useful when available across various computer platforms. An example of this type of document format is the PORTABLE DOCUMENT FORMAT (PDF) developed by Adobe Systems, Incorporated of San Jose, California. XML is in no way this type of document, and Szabo does not in any way suggest such.

For at least the above reasons, all of claims 1-40 are patentable over D'Arlach and Szabo, and these claims should be allowed.

**Claims 6, 7, 21, 22, 24-27, 37 and 38 – Order of Data Import vs. Instruction Performance**

Claims 6, 21, 24 and 37 recite the feature, “the generated instructions further comprise at least one tag indicating an order in which the produced data is to be imported into the electronic document and the instructions are to be performed.” The Advisory Action mailed October 6, 2005 offers in support of the rejection of these claims, “D'Arlach teaches building and/or modifying an HTML [HyperText Markup Language] page. One ordinary skill in the art knows that all HTML pages uses tags.” While this is likely true, the Examiner's argument does not actually address the limitations of the claims, which recite more than just tags. D'Arlach has

been thoroughly reviewed and there is no description or figure in D'Arlach that mentions or shows a tag relating to an order in which to import produced data into an electronic document and to perform instructions that modify the document to accommodate the produced data.

Claims 7, 22, 25-27 and 38 recite specific types of tags that (1) indicate the produced data is to be imported into the electronic document before the instructions are performed, (2) indicate the produced data is to be imported into the electronic document after the instructions are performed, or (3) indicate that at least a portion of the generated instructions are to be inserted into the electronic document. The cited portions of D'Arlach (FIGs. 4-13) are unrelated to this claimed subject matter. The Examiner has failed to even address the previously presented argument regarding these claims. Thus, the record does not support the rejection of claims 7, 22, 25-27 and 38 based on D'Arlach.

For at least the above reasons, claims 6, 7, 21, 22, 24-27, 37 and 38 are patentable over D'Arlach, Szabo and Dilworth, and these claims should be allowed.

#### **Claims 4, 9, 19, 29, 35 and 40 – Retaining Pre-Existing Information and Overlay**

With respect to claims 4, 19, 29 and 35, the cited portion of D'Arlach (col. 5, lines 46-55) describes a server that creates a copy of a Web site template and then modifies that copy in response to user instructions. This does not describe, "adding information to the electronic document without changing pre-existing information in the electronic document", as claimed. To the contrary, a copy of the original template is used in D'Arlach precisely because changes are going to be made that overwrite pre-existing information; thus the pre-existing information is retained in the original document, which is not modified, for later use. The Advisory Action states in response, "D'Arlach teaches that a user can modify certain aspects of the webpage."

This fails to address the actual claim language of claims 4, 19, 29 and 35. Thus, the record does not support the current rejections of claims 4, 19, 29 and 35.

Furthermore, with respect to claims 9 and 40, the cited portions of D'Arlach illustrate a database structure and various user's perspectives of sample display screens. Overlay of visual objects in an electronic document is neither described nor shown in D'Arlach. As described in the present Specification at ¶ 37:

Moreover, additional benefits may be obtained in the context of final format electronic documents by not re-flowing the electronic document in light of the modification. For example, if the electronic document is a form stored in PDF, the instructions may add a new field to the existing form by appending attribute definitions to the PDF file that describe how to construct the field (e.g., type of field, location, dimensions, color, border, etc.), even if the resulting field would overlay some existing content in the visual presentation of the PDF file. By not re-flowing the electronic document, and thus not changing pre-existing final format information in the electronic document, the predefined final format is preserved/respected. The modified electronic document may include many layers of visual presentation information and effectively retains a history of the changes made, which can easily be undone if desired.

(See the Specification at page 11, line 14 to page 12, line 2.) D'Arlach describes generation of new documents by making a copy of a template document and then making changes to this copy by overwriting information in the document. D'Arlach fails to teach or suggest retention of pre-existing information in a document being modified, or overlay of visual objects in an electronic document, to realize the above described advantages.

For at least the above reasons, claims 4, 9, 19, 29, 35 and 40 are patentable over D'Arlach, Szabo and Dilworth, and these claims should be allowed.

### **Ground of Rejection II - Claims 1-40**

#### **Claims 1, 4, 6, 7, 9, 16, 19, 21, 22, 24-27, 29, 32, 35, 37, 38 and 40**

For at least the reasons addressed above under the sub-headings "Independent Claims 1, 16, 24 and 32 – Client-Side Modification", "Claims 6, 7, 21, 22, 24-27, 37 and 38 – Order of Data Import vs. Instruction Performance" and "Claims 4, 9, 19, 29, 35 and 40 – Retaining Pre-Existing Information and Overlay", claims 1-40 are patentable over D'Arlach and Dilworth, and these claims should be allowed.

#### **Independent Claims 1, 16 and 32 – Predetermined Format**

Dilworth describes an on-line editor that "operates as though embedded in a Java-enabled Web browser which enables a user to edit information in a word processor environment. The editor supports most of the HTML [Hypertext Markup Language] formats, including support for graphics. Because the editor in the preferred embodiment is written in the Java programming language, it is cross-platform compatible with most Web browsers. A user is thus required to have a Java-enabled Web browser to access and interactively edit content while on-line to the site. The information generated by the user with the Java editor is stored automatically on the site server when the user clicks on the save button. Thus, there is no need for a site administrator or for any extensive HTML language knowledge, knowledge of Internet protocols, or extensive computer knowledge." (See Dilworth at Abstract.)

The Examiner cites this Abstract of Dilworth as teaching an electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to present the electronic document. This cannot be supported for at least two primary reasons. First, the device-independent aspect of Dilworth is the editor software itself

(due to its being written in the Java programming language) not a predetermined format that defines an appearance of an electronic document as claimed. Second, Dilworth describes a Java editor operating as though embedded in a Web browser to provide a word processor environment to a user, and the editor supports "most of the HTML formats". (*See* Dilworth at page 1, lines 14-17, and page 3, lines 20-25.) Aside from this reference to HTML formats, Dilworth makes no further mention of the format of electronic documents in the system.

The Examiner asserts in the Advisory Action that both the XML and HTML formats are "forms of an electronic document that defines an appearance which is independent of a device", and then goes on to assert that an "XML or HTML page will have the same appearance on any OS system [if a person installed Microsoft Explorer]." This statement is incorrect and is not supported by the record.

An HTML document generally has no ability to specify which portions of the document will appear on which physical pages when printed. In fact, a publisher of an HTML document has no final control over how the document will appear to an end user, because presentation of an HTML document, either by display on a monitor or by printing, is determined by the Web browser, which interprets the HTML tags. Moreover, even if the same Web browser is used to view and print an HTML document on multiple different systems, the HTML document can appear different on the respective systems because Web browsers do not guarantee a consistent appearance for an HTML document across computer platforms. Thus, the reference to HTML formats by Dilworth does not describe the claimed electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to



present the electronic document, and there is no evidence in the record that HTML constitutes this type of electronic document generally.

For at least the above reasons, all of claims 1-40 are patentable over D'Arlach and Dilworth, and these claims should be allowed.

### **Conclusion - Grounds of Rejection I & II - Claims 1-40**

In general, the Examiner's extensive use of bare citations to the figures of D'Arlach without corresponding references to the detailed description is objected to as improper, since the description corresponding to the referenced figures makes clear that D'Arlach neither expressly nor implicitly teaches the presently claimed subject matter. As described in connection with FIG. 4 of D'Arlach:

According to the present embodiment, style templates are stored in the server computer as database files and consist of a set of objects or elements stored in the database. To allow creating a new Web site, a CGI [Common Gateway Interface] program first makes a copy of an existing template in the server computer. The user then customizes or edits the working copy of the template, which is the user's site, through a series of forms displayed by a browser in the client computer. After making desired changes to the site, the user may publish it as a new Web site in case of creating or as a new version of a Web site in case of editing a site. The original templates and their databases are preserved at all times.

A user selects a style template upon which his or her Web site will be based on. In the present embodiment, when a user selects a template to create a site, the template is duplicated in the server computer. All changes to the copy of the template occur on the server computer. When the template is published as a Web site, the database is used to generate a set of Web pages that make up the new site.

(See D'Arlach at col. 5, lines 14-32; emphasis added.) The CGI program is part of the server, and the server interacts with the client in a traditional manner (the server receives requests from the client, and in response, generates a document and delivers such to the client). (See D'Arlach at col. 4, lines 8-51.) The description of FIGs. 5-14 makes clear that a traditional client-server page delivery model is being used (as described in connection with FIGs. 2-3) to enable a user to build a customized Web site on a server. (See D'Arlach at col. 5, line 33 to col. 10, line 3.)

Therefore, D'Arlach clearly describes a system in which a client sends requests to a server, which generates documents to be sent to the client in the course of building a Web site using customizable templates. In every case in D'Arlach, modifications at the server result in the server sending a complete, new HTML document to the client for display in place of a previous HTML document. In contrast, the presently claimed subject matter covers a server that causes a client to receive instructions and data that modify an electronic document at the client. As described in a detailed example in the present application:

The generated instructions are tied to the produced data and to the electronic document in that they are customized to modify the formatting of the electronic document to accept the produced data. By sending the machine instructions with the produced data to the client, the visual presentation formatting of the electronic document may be dynamically changed in a client-server environment to accommodate data produced by a server in response to user activity, without the need to send a new electronic document having visual presentation formatting definitions.

(See ¶ 36 of the Specification at page 11, lines 7-13.) D'Arlach does not teach or suggest client-side modification of electronic documents as recited in the independent claims, to realize the

above described advantages, nor does D'Arlach teach or suggest the details of such modification of an electronic document as recited in the dependent claims.


For all of the above reasons, the rejections of independent claims 1, 16, 24 and 32 as allegedly being unpatentable over D'Arlach, Szabo and Dilworth cannot be maintained, and these claims should thus be patentable over the art of record. Moreover, for at least the reasons stated above, dependent claims 2-15, 17-23, 25-31 and 33-40 should also be patentable over the art of record.

In view of the above, therefore, and with all due respect to the Examiner's position, the rejections of all of claims 1-40 have been respectfully traversed, and the rejections have been proven insufficient to maintain the asserted unpatentability of the claimed subject matter. Thus, reversal of the rejections is respectfully requested.

The brief fee of \$500 is enclosed. Please apply any other necessary charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: December 29, 2005

  
William E. Hunter  
Reg. No. 47,671  
Attorney for Adobe Systems Incorporated

Fish & Richardson P.C.  
Customer Number: 021876  
12390 El Camino Real  
San Diego, California 92130  
Telephone: (858) 678-5070  
Facsimile: (858) 678-5099

### **Appendix of Claims**

1. A machine-implemented method of modifying an electronic document, the method comprising:

receiving a request from a client;

producing data corresponding to the client request;

generating instructions to modify an electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to present the electronic document, the generated instructions specifying one or more operations to modify the electronic document's predetermined final format at the client to accommodate the produced data, the generated instructions to be performed at the client to effect the one or more operations; and

transmitting the produced data and the generated instructions to the client.

2. The method of claim 1, wherein the produced data includes a location from which the client is to retrieve the electronic document.

3. The method of claim 1, wherein the produced data includes a file name for the electronic document, which is already locally accessible by the client.

4. The method of claim 1, wherein the operations to modify the predetermined format at the client comprise adding information to the electronic document without changing pre-existing information in the electronic document.

5. The method of claim 4, wherein the generated instructions comprise a script.

6. The method of claim 5, wherein the generated instructions further comprise at least one tag indicating an order in which the produced data is to be imported into the electronic document and the instructions are to be performed.

7. The method of claim 6, wherein the at least one tag is a before tag, an after tag or a document tag, wherein the before tag indicates that the produced data is to be imported into the electronic document before the instructions are performed, the after tag indicates that the produced data is to be imported into the electronic document after the instructions are performed, and the document tag indicates that at least a portion of the generated instructions are to be inserted into the electronic document.

8. The method of claim 6, wherein the electronic document comprises a form document including one or more form fields, which are responsive to user actions.

9. The method of claim 8, wherein adding the information causes a new visual object to overlay one or more pre-existing visual objects in the electronic document.

10. The method of claim 8, wherein the generated instructions are specific to the electronic document, and the operations to modify the predetermined format at the client result in

one or more of the following document changes: field identity change, field location re-arrangement, and field content change.

11. The method of claim 8, wherein the client comprises a device having a memory storing the electronic document.

12. The method of claim 8, wherein the client comprises a software application.

13. The method of claim 8, wherein the client request comprises a database search request, and wherein producing data comprises retrieving data from a database.

14. The method of claim 8, wherein producing data comprises generating data using scripts.

15. The method of claim 8, wherein generating instructions comprises:  
retrieving initial instructions; and  
customizing the initial instructions to be specific to the electronic document.

16. A machine-implemented method of modifying an electronic document, the method comprising:

obtaining an electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to present the electronic

document;

    sending a request to a server;

    receiving, from the server, data and instructions to modify the electronic document; and

    modifying the electronic document's predetermined final format in accordance with the received instructions to accommodate the received data.

17. The method of claim 16, wherein obtaining the electronic document comprises receiving the electronic document.

18. The method of claim 16, wherein obtaining the electronic document comprises retrieving the electronic document as directed by the instructions.

19. The method of claim 16, wherein modifying the electronic document's predetermined format comprises adding information to the electronic document without changing pre-existing information in the electronic document.

20. The method of claim 19, wherein the instructions comprise a script.

21. The method of claim 20, wherein the instructions further comprise at least one tag indicating an order in which the produced data is to be imported into the electronic document and the instructions are to be performed.

22. The method of claim 21, wherein the at least one tag is a before tag, an after tag or a document tag, wherein the before tag indicates that the produced data is to be imported into the electronic document before the instructions are performed, the after tag indicates that the produced data is to be imported into the electronic document after the instructions are performed, and the document tag indicates that at least a portion of the generated instructions are to be inserted into the electronic document.

23. The method of claim 21, wherein the electronic document comprises a form document including one or more form fields, which are responsive to user actions.

24. A machine-implemented method of modifying an electronic document, the method comprising:

receiving a request from a client;

producing data corresponding to the client request;

generating instructions to modify an electronic document defining visual information to be displayed, the generated instructions specifying one or more operations to modify the electronic document at the client to accommodate the produced data, and the generated instructions including at least one tag indicating an order in which the produced data is to be imported into the electronic document and the instructions are to be performed, and the generated instructions to be performed at the client to effect the one or more operations; and transmitting the produced data and the generated instructions to the client.



25. The method of claim 24, wherein the at least one tag indicates that the produced data is to be imported into the electronic document before the instructions are performed.

26. The method of claim 24, wherein the at least one tag indicates that the produced data is to be imported into the electronic document after the instructions are performed.

27. The method of claim 24, wherein the at least one tag indicates that at least a portion of the generated instructions are to be inserted into the electronic document.

28. The method of claim 24, wherein the at least a portion of the generated instructions result in dynamically set preference settings for the electronic document.

29. The method of claim 24, wherein the operations to modify the electronic document comprise adding formatting information to the electronic document without changing pre-existing formatting information in the electronic document.

30. The method of claim 29, wherein the generated instructions comprise a script.

31. The method of claim 30, wherein the electronic document comprises a form document including one or more form fields, which are responsive to user actions.

32. A machine-readable medium embodying information indicative of instructions for

causing one or more machines to perform operations comprising:

receiving a request from a client;

producing data corresponding to the client request;

generating instructions for an electronic document having a predetermined format that defines an appearance of the electronic document independent of a device used to present the electronic document, the generated instructions specifying one or more operations to modify the predetermined format at the client to accommodate the produced data, the generated instructions to be performed at the client to effect the one or more operations; and

transmitting the produced data and the generated instructions to the client.

33. The machine-readable medium of claim 32, wherein the produced data includes a location from which the client is to retrieve the electronic document.

34. The machine-readable medium of claim 32, wherein the produced data includes a file name for the electronic document, which is already locally accessible by the client.

35. The machine-readable medium of claim 32, wherein the operations to modify the predetermined format at the client comprise adding information to the electronic document without changing pre-existing information in the electronic document.

36. The machine-readable medium of claim 35, wherein the generated instructions comprise a script.

37. The machine-readable medium of claim 36, wherein the generated instructions further comprise at least one tag indicating an order in which the produced data is to be imported into the electronic document and the instructions are to be performed.

38. The machine-readable medium of claim 37, wherein the at least one tag is a before tag, an after tag or a document tag, wherein the before tag indicates that the produced data is to be imported into the electronic document before the instructions are performed, the after tag indicates that the produced data is to be imported into the electronic document after the instructions are performed, and the document tag indicates that at least a portion of the generated instructions are to be inserted into the electronic document.

39. The machine-readable medium of claim 37, wherein the electronic document comprises a form document including one or more form fields, which are responsive to user actions.

40. The machine-readable medium of claim 39, wherein adding the information causes a new visual object to overlay one or more pre-existing visual objects in the electronic document.

Applicant : R. Perelman, et al.  
Serial No. : 10/006,260  
Filed : November 2, 2001  
Page : 28 of 29

Attorney's Docket No.: 07844-501001 / P464

### **Appendix of Evidence**

**NONE**

Applicant : R. Perelman, et al.  
Serial No. : 10/006,260  
Filed : November 2, 2001  
Page : 29 of 29

Attorney's Docket No.: 07844-501001 / P464

**Appendix of Related Proceedings**

**NONE**